Learning temporal matchings for time series discrimination

Cédric Frambourg, Ahlame Douzal-Chouakria, Éric Gaussier

Learning temporal matchings for time series discrimination

Cedric.Frambourg[a,b], Ahlame Douzal-Chouakria[a,*], Eric Gaussier[a]

[a] *Université Joseph Fourier, Grenoble 1 / CNRS / LIG.*
[b] *Université Joseph Fourier, Grenoble 1, France / CNRS / TIMC.*
Cedric.Frambourg,Ahlame.Douzal,Eric.Gaussier@imag.fr

Abstract

In real applications it is not rare for time series of the same class to exhibit dissimilarities in their overall behaviors, or that time series from different classes have slightly similar shapes. To discriminate between such challenging time series, we present a new approach for training discriminative matching that connects time series with respect to the commonly shared features within classes, and the greatest differential across classes. For this, we rely on a variance/covariance criterion to strengthen or weaken matched observations according to the induced variability within and between classes. In this paper, learned discriminative matching is used to define a locally weighted time series metric, which restricts time series comparison to discriminative features. The relevance of the proposed approach is studied through a nearest neighbor time series classification on real datasets. The experiments performed demonstrate the ability of learned matching to capture fine-grained distinctions between time series, and outperform the standard approaches, all the more so that time series behaviors within the same class are complex.

Keywords: time series classification, temporal matching, discriminant analysis, variance-covariance.

1. Introduction

Time series originating from the same sources or measuring the same phenomenon are often noisy and tend to have extremely variable timing of their salient features. To allow time series comparison that accounts for delays, numerous alignment strategies have been proposed, such as those based on Dynamic Time Warping (DTW) [16, 10, 17, 12]. Although these have been used effectively in several domains, classical DTW may be of limited efficiency for time series classification in real applications. In fact, the applied DTW alignment yields a local view, as it is performed in light of a single couple of time series, ignoring all other time series dynamics within and between clusters; furthermore, the alignment process used is achieved regardless of the analysis process

*Corresponding author

(as clustering or classification), weakening its efficiency on complex applications.

To partly overcome these problems, several variants of DTW have been proposed to improve performance in classification or clustering. They mostly aim to more finely estimate the DTW parameters, namely, warping constraints, the time weighting, or the underlying cost function. In Yu et al. [20], without being exhaustive, a multiple bands global path constraint, extending the Sakoe-Chiba band [16], is learned through a brute-force search to maximize the marginal nearest neighbor classification. In Ratanamahatana and Keogh [15], a multiple bands global path is estimated for each class through a forward or backward hill-climbing search to maximize the kNN accuracy. These methods assume equally weighted times and a constant values-based cost function. In Gaudin and Nicoloyannis [6], a genetic algorithm approach is used to approximate a global time weighting matrix. In Jeong et al. [9], a global logistic weight function is evaluated by searching empirically through the entire data set, and aligned values are penalized according to the induced warping. In both works, the time weighting estimation is similarly formulated as a kNN accuracy maximization problem; as in classical DTW, they involve a values-based cost function without warping constraints. In Douzal-Chouakria and Amblard [4], a unified formalism for an adaptive DTW cost function is proposed, specifically to cover both the behavior and value components; the cost function parameters are estimated using a classification tree for time series. In Xie and Wiltgen [19], DTW based on a linear cost function is proposed, involving local (i.e. derivative based) and global (i.e. values-based) features. The latter methods focus on estimating an adaptive cost function while considering equally weighted times without warping constraints.

Using an approach different from DTW alignment, Gaffney et al. [5] propose a probabilistic model that allows for the derivation of an EM learning algorithm that handles clustering and matching processes jointly. For this purpose, a B-spline regression mixture models are proposed for clustering time series data, augmented for the alignment process, with affine transformations for scaling and translation in time. In the same spirit, Ramsay et al. [14] propose a method that learns an alignment function for each time series, parameterized with order one B-spline coefficients. Both of the above proposed alignments remain limited to time series of the same class, which limits the discriminative power of these methods. In Listegarten et al. [11] a hierarchical Bayesian model is proposed, which aligns time series simultaneously across all classes, while detecting and characterizing class-specific differences. The proposed model assumes that the time series from all clusters originate from the same source, i.e. that they share a common global structure with rare differences between clusters. Although these approaches yield more accurate alignments between time series, they assume that time series from the same class share a single global structure.

However, it is indisputable that time series peculiarities may be more complicated in real applications. In particular, it is not unusual that time series from a

same class exhibit differences in overall behaviors, or that time series behaviors exhibit similarities across classes. Consequently, for time series discrimination, it appears important that the time series matching relies on the commonly shared features within the classes and the most differential ones between classes. Such challenging linkages can be achieved by training time series matching within and between classes to localize discriminative features. However, these temporally unconstrained linkages are difficult to reach with alignments that are mainly founded on monotone functions that preserve the temporal order constraints.

With this in mind, we propose a new approach for training time series discriminative matching that highlights class-specific characteristics and differences. The main idea consists of using a discriminant criterion based on variance/covariance to strengthen or weaken links according to their contributions to the variances within and between classes. The variance/covariance measure is used in many approaches, including discriminant analysis, clustering and classification [7, 2, 13, 18]. However, to the best of our knowledge, it has never been investigated for learning temporal matching to discriminate classes of time series. To this end, we propose a new formalization of the classical variance/covariance for a set of time series, as well as for a partition of time series (Section 2). In Section 3, we present a method for training the intra and inter class time series matching, driven by within-class variance minimization and between-class variance maximization. The learned discriminative matching is then used to define a locally weighted time series metric that restricts the time series comparison to discriminative features (Section 4). The relevance of the proposed approach is studied by k-nearest neighbor time series classification on real datasets. In Section 5, the experiments carried out reveal that the proposed approach is able to capture fine-grained distinctions between time series, all the more so that time series of a same class exhibit dissimilar behaviors.

Let us underline the main characteristics of the proposed method: 1) It enlarges time series alignments to a general temporal matching that localizes the common features within classes and the distinctive ones between classes, 2) It takes into account time series of possibly dissimilar behaviors within classes; 3) It is trained according to the temporal dynamics of all time series within and between classes.

2. Variance/covariance for time series

We first recall the definition of the conventional variance/covariance matrix, prior to introducing its formalization for time series data. Let X be the $(n \times p)$ data matrix containing n observations of p numerical variables. The conventional $(p \times p)$ variance/covariance matrix expression is:

$$V = X^t(I - UP)^t P(I - UP)X \quad (1)$$

where, I is the diagonal identity matrix, U the unit matrix, and P a diagonal weight matrix of general term $p_i = \frac{1}{n}$ for equally weighted observations.

In the following, we provide a generalization of the variance/covariance expression Eq.(1) to multivariate time series observations.

2.1. Variance induced by a set of time series

For a set of time series, let X Eq.(2) be the $(nT \times p)$ matrix providing the description of n multivariate time series $S_1, ..., S_n$ by p numerical variables at T time stamps.

$$X = \begin{array}{c} \\ ... \\ S_l \\ ... \end{array} \begin{array}{c} \begin{array}{ccc} X_1 & ... & X_p \end{array} \\ \begin{pmatrix} & ... & \\ x^l_{11} & ... & x^l_{1p} \\ ... & ... & ... \\ x^l_{T1} & ... & x^l_{Tp} \\ & ... & \end{pmatrix} \end{array} \tag{2}$$

The matching between n time series can be described by a matrix M Eq.(3) of positive terms composed of n^2 block matrices $M^{ll'}$ $(l = 1, ..., n; l' = 1, ..., n)$. A block $M^{ll'}$ Eq.(4) is a $(T \times T)$ matrix that specifies the matching between S_l and $S_{l'}$, of general term $m^{ll'}_{ii'} \in [0, 1]$ giving the weight of the link between the observation i of S_l and i' of $S_{l'}$.

$$M = \begin{array}{c} \\ S_1 \\ ... \\ S_n \end{array} \begin{array}{c} \begin{array}{ccc} S_1 & ... & S_n \end{array} \\ \begin{bmatrix} M^{11} & ... & M^{1n} \\ ... & M^{ll'} & ... \\ M^{n1} & ... & M^{nn} \end{bmatrix} \end{array} \tag{3}$$

$$M^{ll'} = \begin{array}{c} \\ S_l \\ \\ \end{array} \begin{array}{c} \begin{array}{ccc} & S_{l'} & \end{array} \\ \begin{bmatrix} m^{ll'}_{11} & ... & m^{ll'}_{1T} \\ ... & m^{ll'}_{ii'} & ... \\ m^{ll'}_{T1} & ... & m^{ll'}_{TT} \end{bmatrix} \end{array} \tag{4}$$

In particular, three basic matchings can be considered:

- A complete linkage connecting all observations of S_l and S'_l, whatever their time stamps, is obtained by setting $\forall\, i, i' \in \{1, ..., T\}$, $m^{ll'}_{ii'} = \frac{1}{T}$, defined by $M^{ll'} = \frac{1}{T} U_T$, U_T being the $(T \times T)$ unit matrix
- The Euclidean alignment connecting observations that occur at the same time is obtained by setting $\forall\, i, i' \in \{1, ..., T\}$, $m^{ll'}_{ii'} = 1$ if $i = i'$ and 0 otherwise, described by $M^{ll'} = I$;

- A dynamic time warping alignment is obtained by setting $\forall\, i, i' \in \{1, ..., T\}$, $m_{ii'}^{ll'} = 1$ if i is aligned with i' by the standard DTW, and 0 otherwise.

Then, the $(p \times p)$ variance/covariance matrix V_M induced by a set of time series $S_1, ..., S_n$ connected to one another according to the matching matrix M can be defined on the basis of Eq.(1), as:

$$V_M \quad = \quad X^t (I - M)^t P (I - M) X \tag{5}$$

where P is a $(nT \times nT)$ diagonal matrix of weights, with $p_i = \frac{1}{n\,T}$ for equally weighted observations. Note that for a complete linkage matching, M is equal to UP and V_M leads to a conventional variance covariance V Eq.(1).

For clarity and to simplify notation, we focus for the theoretical developments on univariate time series. The extension to the multivariate case is direct and will be used in the experiments.

Thus, let x_i^l be the value of the variable X taken by S_l $(l = 1, ..., n)$ at the ith time stamp $(i = 1, ..., T)$.

Definition 1. *The variance V_M of the variable X is given by:*

$$V_M \quad = \quad \sum_{l=1}^{n} \sum_{i=1}^{T} p_i (x_i^l - \sum_{l'=1}^{n} \sum_{i'=1}^{T} m_{i\,i'}^{l\,l'} x_{i'}^{l'})^2 \tag{6}$$

Note that each value x_i^l is centered relative to the term $\sum_{l'=1}^{n} \sum_{i'=1}^{T} m_{i\,i'}^{l\,l'} x_{i'}^{l'}$ estimating the average of X in the neighborhood of the time i of S_l. The neighborhood of i is the set of instants i' of $S_{l'}$ $(l' = 1...n)$ connected to i with $m_{ii'}^{ll'} \neq 0$. We now proceed to define the variance within and between classes when the set of time series is partitioned into classes.

2.2. Variance induced by a partition of time series

Let us now consider a set of time series $S_1, ..., S_n$ partitioned into K classes, with $y_i \in \{1, ..., K\}$ the class label of S_i and n_k the number of time series belonging to class C_k. The definition of the *within variance* (i.e. the variance within classes) and the *between variance* (i.e. the variance between classes) induced by K classes is obtained by using the expression given in Eq.(5) based on a matching M specified below.

Definition 2. *The* **within variance** *with an intra-class matching matrix M is given by:*

$$WV_M = \frac{1}{nT} \sum_{k=1}^{K} \sum_{l=1}^{n_k} \sum_{i=1}^{T} (x_i^l - \sum_{l'=1}^{n_k} \sum_{i'=1}^{T} m_{ii'}^{ll'} x_{i'}^{l'})^2$$

with

$$M^{ll'} = \begin{cases} \mathbf{I} & \text{if } l = l' \\ \neq \mathbf{0} & \text{if } y_l = y_{l'} \text{ and } l \neq l' \\ \mathbf{0} & \text{if } y_l \neq y_{l'} \end{cases} \quad (7)$$

where $\mathbf{I}$ *and* $\mathbf{0}$ *are the* $(T \times T)$ *identity and zero matrices, respectively.*

The general setting for the blocks $M^{ll'}$ of the intra-class matching M is based on three considerations: (a) the Euclidean alignment ($M^{ll} = \mathbf{I}$) linking each time series to itself ensures a variance of zero when comparing a time series with itself, (b) time series within the same class should be connected, while (c) time series of different classes are not connected, as they do not contribute to the within variance.

Similarly, we have:

Definition 3. *The* **between variance** *with an inter-class matching matrix* M *is given by:*

$$BV_M = \frac{1}{nT} \sum_{k=1}^{K} \sum_{l=1}^{n_k} \sum_{i=1}^{T} (x_i^l - (m_{ii}^{ll} x_i^l + \sum_{k' \neq k} \sum_{l'=1}^{n_{k'}} \sum_{i'=1}^{T} m_{ii'}^{ll'} x_{i'}^{l'}))^2$$

with

$$M^{ll'} = \begin{cases} \mathbf{I} & \text{if } l = l' \\ \mathbf{0} & \text{if } y_l = y_{l'} \text{ and } l \neq l' \\ \neq \mathbf{0} & \text{if } y_l \neq y_{l'} \end{cases} \quad (8)$$

where $\mathbf{I}$ *and* $\mathbf{0}$ *are the* $(T \times T)$ *identity and zero matrices, respectively.*

The setting of the inter-class matching M is symmetric with respect to the preceding one, matching between time series of the same class being forbidden, while matching between time series of different classes is taken into account.

As one can note, the matching matrix M plays a crucial role in the definition of the within and between variances. The main issue for time series classification is therefore to learn a discriminative matching that highlights shared features within classes and distinctive ones between classes. To do so, we look for the matching matrix M, under the general settings given in Eqs. (7) and (8), that minimizes the within variance and maximizes the between variance. We present an efficient way to do this in the following section.

3. Learning discriminative matchings

We present here an efficient method to learn the matching matrix M, so as to connect time series based on their discriminative features. The proposed approach consists of two successive phases. In the first phase, the intra-class matching is learned to minimize the within variance. The learned intra-class matching reveals time series connections based on class-specific characteristics. In the second phase, the learned intra-class matching is refined to maximize the between variance.

3.1. Learning the intra-class matching

We are interested in inferring commonly shared structure within classes, that is in identifying the set of time stamps i' connected to each time stamp i regardless of their weights.

Thus, the problem of learning the intra-class matching matrix M to minimize the within variance, i.e. the quantity WV_M of Definition 2, can be formulated as the following constrained optimization problem:

$$\begin{cases} \arg\min_M & WV_M \\ \text{subject to:} & \forall k \in \{1, ..., K\},\ \forall (l, l') \in C_k,\ \forall (i, i') \in [1, T]^2 : \\ & m_{ii}^{ll} > 0 \text{ and } m_{ii'}^{ll} = 0 \text{ for } i \neq i' \\ & \sum_{i'=1}^{T} m_{ii'}^{ll'} > 0, \\ & \sum_{l'=1}^{n_k} \sum_{i'=1}^{T} m_{ii'}^{ll'} = 1 \\ & \text{if } m_{ii'}^{ll'} \neq 0,\ m_{ii'}^{ll'} = m_{ii}^{ll} \end{cases} \tag{9}$$

The first three constraints are dictated by the variance/covariance definition. More precisely, the first constraint ensues from the first setting ($M^{ll} = \mathbf{I}$) given in Eq. (7). The second constraint guarantees the second setting ($M^{ll'} \neq \mathbf{0}$) by imposing that each time stamp i of S_l be connected to at least one time i' of $S_{l'}$. The third one corresponds to a row normalization of M involved in the centering process of WV_M (Definition 2). The last constraint determines the linkage structure to be extracted, namely, equally weighted links in the neighborhood of i, as the interest is to reveal time stamps i' connected to i, regardless of their weights.

The fourth constraint renders the problem discrete and standard gradient approaches are inappropriate. Furthermore, an exhaustive search is in practice unfeasible, because the number of configurations is $(2^T - 1)^{\sum_k (n_k - 1) n_k T} \simeq 2^{\sum_k n_k^2 T^2}$. We introduce here an efficient approach that iteratively evaluates the contribution of each linked observation (i, i') to the within variance; the weights $m_{ii'}^{ll'}$ are then penalized for all links (i, i') that significantly increase the within variance. This process, called *TrainIntraMatch*, is described in Algorithm 1 and involves the following steps.

Step1: Initialization A complete linkage is used to initialize the intra-class matching matrix M, to ensure that all possible matchings are considered and that no *a priori* constraints on the type of matching one should look for are introduced. Furthermore, it satisfies the constraints given in Eq.(9).

$$M^{ll'} = \begin{cases} \mathbf{I} & \text{if } l = l' \\ \frac{1}{T}\mathbf{U} & \text{if } y_l = y_{l'} \text{ and } l \neq l' \\ \mathbf{0} & \text{if } y_l \neq y_{l'} \end{cases} \tag{10}$$

Algorithm 1 *TrainIntraMatch*(X, α)

M = complete intra-class matching — Step 1
repeat
  $LinkRemoved = false$
  for all (l, l') *with* $y_l = y_{l'}$ and $l \neq l'$ **do**
    for all $(i, i') \in [1, T] \times [1, T]$ **do**
      $C_{ii'}^{ll'}$ evaluation with Eq. (11) — Step 2
    end for
  end for
  for all $(i, l) \in [1, T] \times [1, n]$ **do**
    $Link = \arg\max_{i',l'}(C_{ii'}^{ll'})$ satisfying Eq. (13) — Step 3
    if $Link \neq \emptyset$ **then**
      Remove $Link$ ($m_{i,i'}^{l,l'} = 0$) and
      Update weights with Eq. (12)
      $LinkRemoved = true$
    end if
  end for
until $\neg LinkRemoved$ — Step 4
$return(M_{Intra} = M)$

Step 2: Computing link contributions We define the contribution $C_{i_1 i_2}^{l_1 l_2}$ of the link (i_1, i_2) between S_{l_1} and S_{l_2} $(y_{l_1} = y_{l_2})$ as the induced variation on the within variance after the link (i_1, i_2) has been removed:

$$C_{i_1 i_2}^{l_1 l_2} \quad = \quad WV_M - WV_{M \backslash (i_1, i_2, l_1, l_2)} \tag{11}$$

where $M\backslash(i_1, i_2, l_1, l_2)$ denotes the matrix obtained from M by setting $m_{i_1 i_2}^{l_1 l_2}$ to 0 and re-normalizing its i_1^{th} row:

$$m_{i_1 i'}^{l_1 l'} \leftarrow \frac{m_{i_1 i'}^{l_1 l'}}{1 - m_{i_1 i_2}^{l_1 l_2}} \tag{12}$$

The evaluated contributions reveal two types of links: the links of positive contribution $C_{ii'}^{ll'} > 0$ that decrease the within variance if removed, and the links of negative contribution $C_{ii'}^{ll'} < 0$ that increase the within variance if removed.

Step 3: Link deletion The deletion of a link with positive contribution ensures that the within variance will decrease. Because of the renormalization given in Eq.(12), the second and third constraints given in Eq.(9) are satisfied. However, one should not remove a link if its deletion violates the fourth constraint. In addition, if all links within a row have a negligible contribution to the variance, one can dispense with removing them in order to (a) avoid overtraining and (b) speed up the process. Thus, a link (i, i') between S_l and $S_{l'}$ is deleted if it satisfies:

$$C_{ii'}^{ll'} > \alpha . WV_{M_1} \quad \text{and} \quad \sum_{i''=1, (i'' \neq i')}^{T} m_{ii''}^{ll'} > 0 \tag{13}$$

where $\alpha \in [0,1]$ and WV_{M_1} is the initial within variance. When $\alpha = 0$, all links with positive contributions are deleted as long as this deletion does not violate the constraints.

Because the normalization in Eq.(12) performed after the deletion of (i_1, i_2) impacts only the weights of the i_1^{th} row, deleting a single link per row at each iteration of the process guarantees that the global within variance will decrease. Thus, at each iteration one can simply delete the link on each row of maximal contribution compliant with Eq.(13).

Step 4: Stopping the learning process. The algorithm iterates steps 2, 3 and 4 until there are no more links satisfying the conditions specified in Eq.(13).

From the learned intra-class matching obtained at step 4, noted M_{Intra}, one may induce for each time series S_l one intra-block $M^{l.}_{Intra}$ to indicate the characteristic linkage between S_l and time series of the same class. This intra-block is obtained by summing the block matrices learned for S_l, as follows:

$$M^{l.}_{Intra} = \sum_{l' \in 1,\ldots,n_k} M^{ll'}_{Intra} \tag{14}$$

Note that the row normalization of M_{Intra} assures the normalization of $M^{l.}_{Intra}$. Furthermore, post-pruning can be carried out on $M^{l.}_{Intra}$ by setting to 0 all the weights lower than the initial uniform weighting (weight $< \frac{1}{T}$), assumed not significant for classes characterization.

3.2. Learning the inter-class matching

The goal of this second phase is to refine the highlighted connections in M_{Intra} (i.e., that connects shared features within classes) to capture the links that are additionally differentiating classes. For this, we refer to a similar algorithm called *TrainInterMatch*, where the inter-class matching is initialized with M_{Intra}, then trained to maximize the between variance. Similarly, the problem of learning the inter-class matching matrix M to maximize the between variance, i.e. the quantity BV_M of Definition 3, can be formulated as the following constrained optimization problem:

$$\begin{cases} \arg\max_M & BV_M \\ \text{subject to:} & \forall\, k \in \{1,...,K\},\, \forall\, l \in C_k,\, \forall\, l' \notin C_k,\, \forall\, (i,i') \in [1,T]^2 : \\ & m^{ll}_{ii} > 0 \text{ and } m^{ll}_{ii'} = 0 \text{ for } i \neq i' \\ & \sum_{i'=1}^{T} m^{ll'}_{ii'} > 0, \\ & m^{ll}_{ii} + \sum_{k' \neq k} \sum_{l'=1}^{n'_k} \sum_{i'=1}^{T} m^{ll'}_{ii'} = 1 \\ & \text{if } m^{ll'}_{ii'} \neq 0,\, m^{ll'}_{ii'} = m^{ll}_{ii} \end{cases} \tag{15}$$

The first constraint derives from the first setting ($M^{ll} = \mathbf{I}$) given in Eq. (8). The second constraint guarantees the third setting ($M^{ll'} \neq \mathbf{0}$) of Eq. (8) by

imposing that each time stamp i of S_l should be connected to at least one time i' of $S_{l'}$. The third one corresponds to row normalization of M involved in the centering process of BV_M (Definition 3), and the last condition ensures equally weighted links in the neighborhood of i.

As for the within variance minimization problem, the fourth constraint makes the problem discrete and standard gradient approaches are inappropriate. In addition, an exhaustive search is unfeasible as the number of configurations is $2^{n^2T^2}$. Thus, we adopt the same approach, which consists in iteratively evaluating the contribution of each linked observations (i, i') to the between variance; the weights $m_{ii'}^{ll'}$ are then penalized for all links (i, i') significantly decreasing the between variance. We now briefly describe the main steps of the *TrainInterMatch* algorithm 2.

Algorithm 2 *TrainInterMatch*(X, α)

M = defined from M_{Intra} with Eq. (16) — Step 1
repeat
  $LinkRemoved = false$
  for all (l, l') *with* $y_l \neq y_{l'}$ **do**
    for all $(i, i') \in [1, T] \times [1, T]$ **do**
      $C_{ii'}^{ll'}$ evaluation with Eq. (17) — Step 2
    end for
  end for
  for all $(i, l) \in [1, T] \times [1, n]$ **do**
    $Link = \arg\min_{i',l'}(C_{ii'}^{ll'})$ satisfying Eq. (19) — Step 3
    if $Link \neq \emptyset$ **then**
      Remove $Link$ ($m_{i,i'}^{l,l'} = 0$) and
      Update weights with Eq. (18)
      $LinkRemoved = true$
    end if
  end for
until $\neg LinkRemoved$ — Step 4
$return(M_{Intra} = M)$

Step1: Initialization The inter-class matching matrix M is initialized as follows:

$$M^{ll'} = \begin{cases} \mathbf{I} & \text{if } l = l' \\ \mathbf{0} & \text{if } y_l = y_{l'} \text{ and } l \neq l' \\ M_{Intra}^{l.} & \text{if } y_l = k \neq y_{l'} \end{cases} \quad (16)$$

where $M^{ll'}$ is initialized with the aggregated intra block $M_{Intra}^{l.}$ of S_l according to Eq.(14). The row normalized initial matrix is in agreement with the constraints given in Eq.(15).

Step 2: Computing link contributions We define the contribution $C_{i_1 i_2}^{l_1 l_2}$ of the link (i_1, i_2) between S_{l_1} and S_{l_2} $(y_{l_1} \neq y_{l_2})$ to be the variation induced on the between variance after the link (i_1, i_2) has been removed:

$$C_{i_1 i_2}^{l_1 l_2} \quad = \quad BV_M - BV_{M\setminus(i_1,i_2,l_1,l_2)} \tag{17}$$

where $M\setminus(i_1, i_2, l_1, l_2)$ denotes the matrix obtained from M by setting $m_{i_1 i_2}^{l_1 l_2}$ to 0 and re-normalizing its i_1^{th} row:

$$m_{i_1 i'}^{l_1 l'} \leftarrow \frac{m_{i_1 i'}^{l_1 l'}}{1 - m_{i_1 i_2}^{l_1 l_2}} \tag{18}$$

The evaluated contributions reveal two types of links: those making a positive contribution $C_{ii'}^{ll'} > 0$ that induce a decrease of the between variance if removed and those making a negative contribution $C_{ii'}^{ll'} < 0$ that cause an increase of the between variance if removed.

Step 3: Link deletion The deletion of a link with a negative contribution ensures that the between variance will increase. The second and third constraints given in Eq.(15) are preserved by the row normalization given in Eq.(18). However, one should not remove a link if its deletion would violate the fourth constraint. In addition, if all one can dispense with removing links with negligible contributions in order to (a) avoid overtraining and (b) speed up the process. Thus, a link (i, i') between S_l and $S_{l'}$ is deleted if it satisfies:

$$C_{ii'}^{ll'} < -\alpha.BV_{M_1} \text{ and } \sum_{i''=1,(i''\neq i')}^{T} m_{ii''}^{ll'} > 0 \tag{19}$$

where $\alpha \in [0, 1]$ and BV_{M_1} is the initial between variance. When $\alpha = 0$, all links with negative contributions are deleted as long as this deletion does not violate the constraints. As argued before, the convergence of the learning process is ensured by removing, at each iteration, one link at most for each row of M, the one exhibiting the minimal negative contribution compliant with Eq.(19).

Step 4: Stopping the learning process. The algorithm iterates steps 2, 3 and 4 until there are no more links satisfying the conditions specified in Eq.(19). Let M_* be the learned intra-class matching.

3.3. Convergence and complexity of the learning process

As noted above, the process retained guarantees that each time one deletes a link compliant with Eq.(13) in *TrainIntraMatch* (respectively to Eq.(19) in *TrainInterMatch*) and renormalizes M according to Eq.(12) (respectively Eq.(18)), the variance decreases (respectively increases) while the constraints are still satisfied. This process thus converges towards a matching matrix M, yielding a lower within class (respectively higher between class) variance than the original

one. Furthermore, if α equals 0, the set of links obtained is minimal in the sense that any deletion of a link from this set will lead to an increase (respectively decrease) in the variance.

In terms of complexity, the dominating factor in the above process is the computation of the contributions of each link. This contribution $C_{i_1 i_2}^{l_1 l_2}$ can be expressed as the difference between the centered values of $x_{i_1}^{l_1}$ before and after the deletion of (i_1, i_2):

$$\begin{aligned} C_{i_1 i_2}^{l_1 l_2} &= \frac{1}{nT}(x_{i_1}^{l_1} - \sum_{l'=1}^{n_k}\sum_{i'=1}^{T} m_{i_1 i'}^{l_1 l'} x_{i'}^{l'})^2 - \\ & \quad \frac{1}{nT}(x_{i_1}^{l_1} - \sum_{l'=1}^{n_k}\sum_{i'=1}^{T} \frac{m_{i_1 i'}^{l_1 l'}}{1 - m_{i_1 i_2}^{l_1 l_2}} x_{i'}^{l'})^2 \end{aligned} \quad (20)$$

for the *TrainIntraMatch*, and:

$$\begin{aligned} C_{i_1 i_2}^{l_1 l_2} &= \frac{1}{nT}(x_{i_1}^{l_1} - (m_{ii}^{ll} x_i^l + \sum_{k' \neq k}\sum_{l'=1}^{n_{k'}}\sum_{i'=1}^{T} m_{ii'}^{ll'} x_{i'}^{l'}))^2 - \\ & \quad \frac{1}{nT}(x_{i_1}^{l_1} - (\frac{m_{i_1 i_1}^{l_1 l_1}}{1 - m_{i_1 i_2}^{l_1 l_2}} x_i^l + \sum_{k' \neq k}\sum_{l'=1}^{n_{k'}}\sum_{i'=1}^{T} \frac{m_{i_1 i'}^{l_1 l'}}{1 - m_{i_1 i_2}^{l_1 l_2}} x_{i'}^{l'}))^2 \end{aligned} \quad (21)$$

for the *TrainInterMatch*.

The complexity of the *TrainIntraMatch* algorithm thus amounts to $O(\sum_{k=1}^{K} I_k n_k^2 T^2)$ where I_k corresponds to the number of times the process has affected elements of class k. In the worst case, for each class k, $I_k = (n_k - 1)(T - 1)$, and the overall complexity amounts to $O(\sum_{k=1}^{K} n_k^3 T^3)$. Similarly, the complexity of the *TrainInterMatch* algorithm is about $O(\sum_{k=1}^{K} I_k (n - n_k) n_k T^2)$ with, in the worst case, $I_k = (n - n_k)(T - 1)$, and an overall complexity of $O(\sum_{k=1}^{K} (n - n_k)^2 n_k T^3) \simeq O(n^3 T^3)$.

We now turn to the application of the learned matching matrix to time series classification.

4. Time series classification based on the learned matching

Our aim here is to present a way of using learned discriminative matching to locally weight time series for k-nearest neighbor classification. The purpose of the proposed weighting is to restrict the time series comparison to the discriminant (characteristic and differential) features. Let M_* be the discriminative matching learned by the *TrainIntraMatch* and *TrainInterMatch* algorithms, where discriminant linkages are highly weighted. For each S_l of the training sample, we define its discriminative matching $M_*^{l\cdot}$ as the average of the learned matrices $M_*^{ll'}$ ($y_{l'} \neq y_l = k$):

$$M_*^{l\cdot} = \frac{1}{(n-n_k)\,T}\sum_{l'} M_*^{ll'}$$

In k-nearest neighbor classification, one can compare a new time series S_{test} to a sample series S_l of C_k based on its learned discriminative matching. However, as discussed above, discriminative features of time series of C_k may appear at different time stamps, according to a delay r. Thus, to evaluate the proximity to S_l one should consider, in addition, the delay inherent in S_{test}. This can be achieved by looking for the delay r that leads to the minimal distance between S_{test} and S_l, as proposed in the following locally weighted proximity measure:

$$D_l(S_l, S_{test}) = \min_{r\in\{0,..,T-1\}}\Big(\sum_{|i-i'|\leq r;\,(i,i')\in[1,T]^2} \frac{m_{ii'}^{l\cdot}}{\sum_{|i-i'|\leq r} m_{ii'}^{l\cdot}}(x_i^l - x_{i'}^{test})^2\Big) \qquad (22)$$

where r corresponds to the Sakoe-Chiba band width [16]. Note that for $r = 0$, D_l defines a locally weighted Euclidean distance involving the diagonal weights $m_{ii}^{l\cdot}$, whereas for $r = T-1$ the time series are compared according to the complete learned discriminative matching. The proposed D_l defines a dissimilarity index satisfying the positivity, symmetry and coincidence axioms. The D_l is simply noted D in the following.

5. Experiments

5.1. Description of the datasets

To motivate our approach we first considered BME and UMD, two synthetic challenging datasets composed of time series that are dissimilar within classes and slightly similar between classes. BME consists of three classes *Begin*, *Middle*, and *End* (Figure 1). In the *Begin* (respectively the *End*) class, the time series share a common signature defined by a small bell arising at the initial (respectively final) period. The overall behavior may be distinctive within these classes depending on whether the large bell is up or down positioned. Furthermore, time series of the *Begin* and the *End* classes composed of an up-positioned large bell are quite similar to the *Middle* class time series.

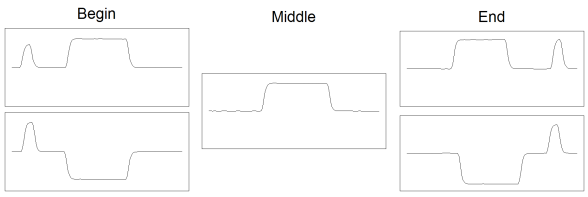

Figure 1: Distinctive behaviors within BME classes: *Begin*, *Middle*, and *End*

The second dataset UMD, composed similarly of three classes *Up*, *Middle*, and *Down*, introduces more complexity, with a local shared signature (i.e. a

small bell) occurring at different time stamps, as illustrated in Figure 2.

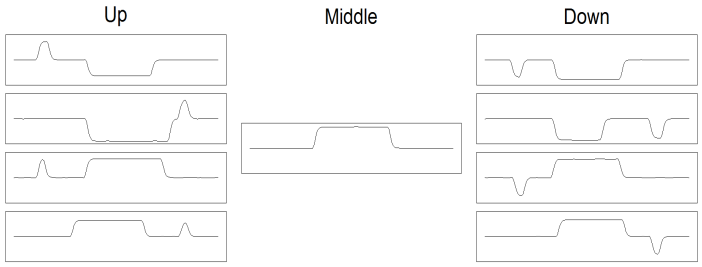

Figure 2: Distinctive behaviors within UMD classes: UP, MIDDLE, and DOWN

The reason for considering these synthetic datasets is to show, through a challenging but easily identifiable discriminant features, which connections are selected by the *TrainIntraMatch* algorithm and how they are refined after the *TrainInterMatch* algorithm captures the discriminative linkage.

The proposed approach is thus motivated by a real application that aims to analyze the electrical power consumption of customers, to adequately meet consumer demands. We use two datasets CONSLEVEL and CONSSEASON obtained from a public database[1] providing the electric power consumption recorded in a personal home over almost one year (349 days). Each time series consists of 144 measurements that give the power consumption of one day with a 10 minute sampling rate.

CONSLEVEL divides the 349 time series into two classes (*Low* and *High*) depending on whether the average electric power during the peak demand period (6:00pm-8:00pm) is lower or greater than the annual average consumption during that period. Figure 3 shows the pattern of electric consumption within the CONSLEVEL classes; the red frames delineate the time interval [108,120], which corresponds to the peak period (6:00pm-8:00pm).

On the other hand, CONSSEASON splits the 349 time series into two season classes (*Warm* and *Cold*) depending on whether the power consumption is recorded during the warm (from April to September) or cold (from October to March) seasons (Figure 4). Note that the electric power consumption profiles differ markedly within classes in both datasets.

The goal of the proposed approach applied to the CONSLEVEL and CONSSEASON datasets is: to localize the periods that characterize the daily power

[1]These data are available at http://bilab.enst.fr/wakka.php?wiki=HomeLoadCurve, and analyzed in [8]

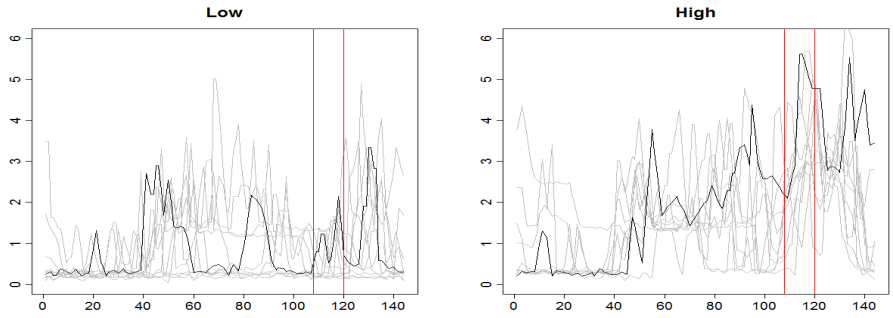

Figure 3: The electrical power consumption within the *Low* and *High* classes of CONSLEVEL dataset.

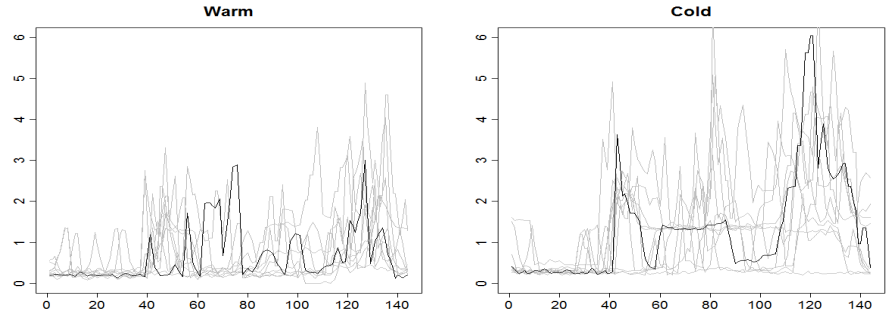

Figure 4: The electrical power consumption within the *Warm* and *Cold* classes of CONSSEASON dataset.

consumption of each class, to highlight periods that differentiate the power consumption of different classes, and to classify new power consumption based on the learned discriminative features. For instance, for the CONSLEVEL dataset, a classification based on the discriminative periods prior to the time interval 6:00pm-8:00pm can help forecast the consumer demands during the peak period.

In addition to the above mentioned datasets, we have used a standard dataset on character trajectories TRAJ [1], where time series share a quite similar global behavior within classes (20 classes of 50 time series each). The goal of this latter dataset is to verify whether the proposed approach can recover standard time series structures within classes or not.

5.2. Results and discussion

The algorithms *TrainIntraMatch* and *TrainInterMatch* are applied to the above datasets with $\alpha = 0.5\%$. As an example, let us first illustrate, for the

BME dataset, the progression of the within and between variances during the learning processes (Figure 5). The clearly monotonically decreasing (respectively increasing) behavior of the within (respectively between) class variance, which ends at a plateau, assesses: a) the pertinence of the conducted links penalization to minimize the within variance and maximize the between variance, b) the convergence of the proposed algorithms.

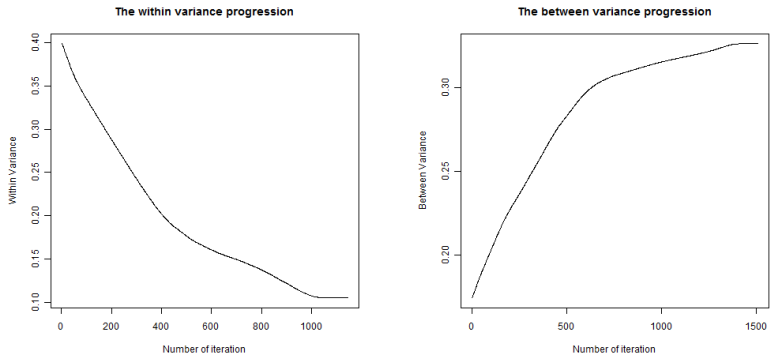

Figure 5: The within and between variance progression for BME dataset.

Figure 6 displays, for a time series from the *Middle* class (UMD dataset), its learned intra-class and inter-class blocks; the bright cells indicate highly weighted links. The intra-class block (Figure 6 left) reveals the characteristic matching between the given time series (in a row) and time series of the same class (in a column). The determined structure, from the intra-class block, shows a strong linkage between, on the one hand, the central large bells represented by the central light square, and on the other hand, the initial and final plateaus. From the corresponding inter-block (Figure 6 right), we can see that connections that are characteristic (captured in the intra-class block) but not differential have been removed, while those that are both characteristic and differential are retained and reinforced. In particular, note that the weights of the central square region are lowered but not set to zero, because this region discriminates time series of the *Up* and *Down* classes that are composed of a down positioned large bell. However, the linkage between the initial and final plateaus has been drastically reduced to small discriminative regions (i.e., highly weighted), corresponding to the periods where the small bells may arise within the *Up* and *Down* classes.

For CONSLEVEL, similarly, we can see in Figure 7 the learned intra-class and inter-class blocks for a given time series of the *Low* class. The intra-class block reveals a checkerboard structure, indicating that the electric power consumption within the *Low* class alternates, in a daily period, between a low and a moderately high consumption. The corresponding inter-class block shows the

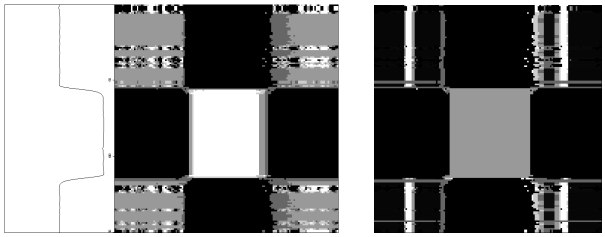

Figure 6: The learned intra (left) and inter (right) class matching for UMD dataset.

discriminative matching between the considered *Low* class time series and time series of the *High* class (on column). This block displays many discriminative regions; for example, it shows that the power consumption within the *High* class within the period underlined in red (prior to 6:00pm-8:00pm) is especially important in predicting the consumption during the peak period.

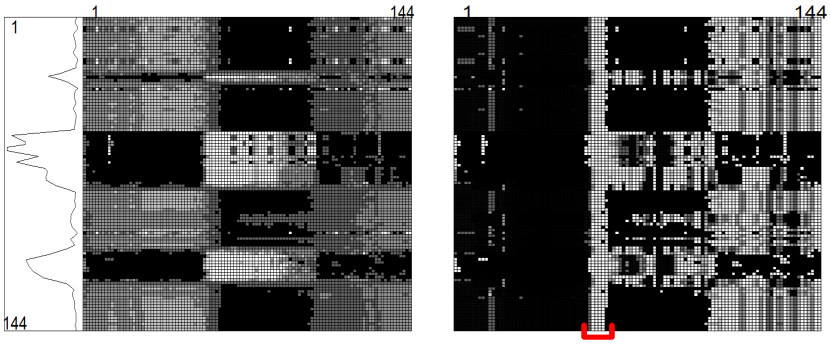

Figure 7: The learned intra and inter class matching for a *Low* class time series.

The learned discriminative matching is then used for the locally weighted time series metric D given in Eq.(22). The relevance of the proposed approach and of the induced metric are then studied through a k-nearest neighbor classification for $k = 1, 3, 5, 7$ and through a leave-one-out protocol. The results obtained are compared to two baselines: the Euclidean DE and dynamic time warping DTW distances (Table 1).

The misclassification error rates obtained in Table 1 show the efficiency of the proposed locally weighted metric D in discriminating between complex time series classes, compared to standard metrics for time series. In particular, one can note that for all datasets but TRAJ, the best results (in bold) are obtained with D. For TRAJ, the three metrics lead to comparable results suggesting that the Euclidean alignment is an appropriate matching for this dataset. In Figure 8, we can see that the learned discriminative matching, for example, for "c", "o", "i", "e", "u" and "a" characters is close to the Euclidean one, which shows

Table 1: k-Nearest Neighbor classification error rates

	k	D	DE	DTW
BME	1	**0.032**	0.165	0.130
	3	0.034	0.208	0.132
	5	0.062	0.234	0.136
	7	0.079	0.297	0.191
UMD	1	**0.055**	0.173	0.121
	3	0.111	0.333	0.177
	5	0.173	0.343	0.225
	7	0.222	0.378	0.274
CONSLEVEL	1	0.056	0.306	0.289
	3	0.044	0.267	0.261
	5	0.028	0.233	0.239
	7	**0.017**	0.233	0.233
CONSSEASON	1	**0.094**	0.239	0.283
	3	0.128	0.228	0.311
	5	0.205	0.200	0.300
	7	0.111	0.222	0.306
TRAJ	1	0.014	**0.012**	0.019
	3	0.018	0.017	0.022
	5	0.022	0.021	0.028
	7	0.019	0.021	0.026

the ability of the proposed approach to recover standard time series alignments. In addition, one can see that for nearly all datasets the best performances are obtained for $k = 1$. For CONSLEVEL, a slight improvement is reached for $k = 7$, indicating a great clusters overlap for this dataset.

Finally, to complete the above observed results and to assess the discriminative strength of the proposed metric with respect to the baselines retained, classical multidimensional-scaling [3] is used to visualize time series proximities induced by the studied metrics for both CONSLEVEL and CONSSEASON datasets (Figure 9). These figures corroborate the discriminative power of the proposed metric, because they display compact and well isolated classes, whereas baselines present a great overlap between the different classes.

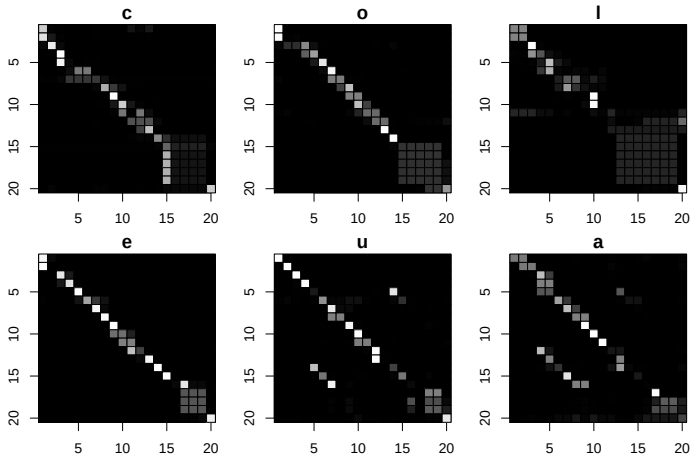

Figure 8: The learned discriminative matching for the characters "c", "o", "l", "e", "u" and "a" of TRAJ dataset.

6. Conclusion

Motivated by discriminating time series that are dissimilar within classes or nearly similar across classes, we have presented a new approach for training discriminative matchings that connects time series with respect to the commonly shared features within classes, and the most differential ones across classes. To do so, we have first introduced a generalization of the variance to sets of time series. The definition we have provided generalizes the standard definition of the variance and is the first one proposed, to our knowledge, for sets of time series. We have then introduced efficient approaches to learn matching matrices between time series that minimize (respectively maximize) the within (respectively between) class variance. Based on the learned matching, we have finally introduced a new locally weighted metric that restricts the time series comparison to discriminative features. The experiments we have conducted show the ability of the learned matching to capture fine-grained distinctions between time series; they also show that the metric we have introduced outperforms metrics commonly used on time series, as the Euclidean and Dynamic Time Warping distances, on two real datasets.

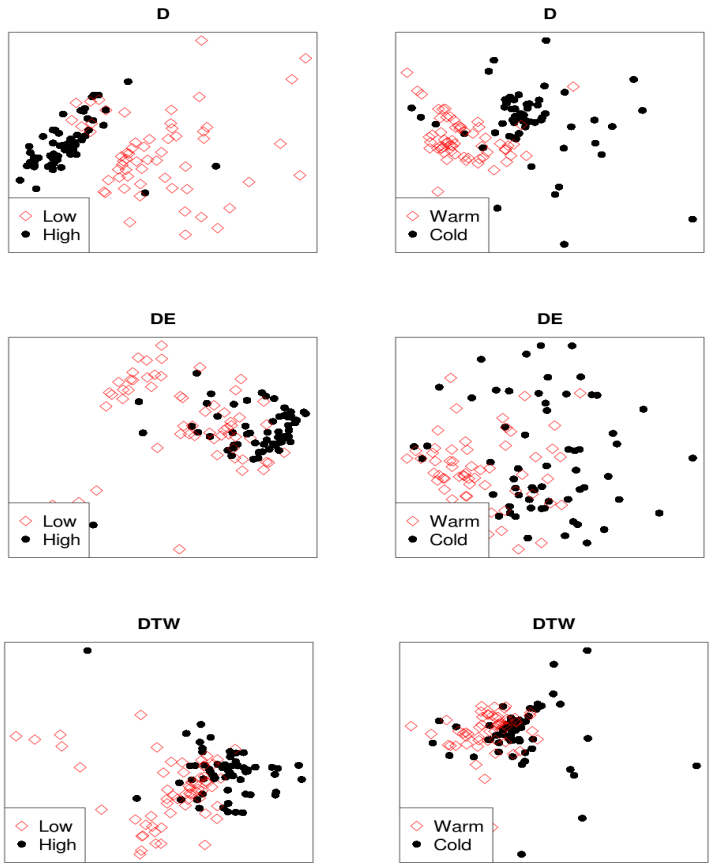

Figure 9: The time series proximities induced by D, DE and DTW for CONSLEVEL and CONSSEASON data.

References

[1] Asuncion, A., & Newman, D. (2007). Uci machine learning repository.

[2] Chopra, S., Hadsell, R., & LeCun, Y. (CA, 2005). Learning a similiarty metric discriminatively, with application to face verification. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 349–356). San Diego.

[3] Cox, T., & Cox, M. (2001). *Multidimensional Scaling*. Chapman and Hall.

[4] Douzal-Chouakria, A., & Amblard, C. (2012). Classification trees for time series. *Pattern Recognition*, *45*, 1076–1091.

[5] Gaffney, S. J., & Smyth, P. (2005). Joint probabilistic curve clustering and alignment. *In Advances in Neural Information Processing Systems*, *17*, 473–480.

[6] Gaudin, R., & Nicoloyannis, N. (2006). An adaptable time warping distance for time series learning. In *the 5th International Conference on Machine Learning and Applictions.* (pp. 213–218).

[7] Hastie, T., & Tibshirani, R. (1996). Discriminant adaptive nearest neighbor classification. *IEEE Transacations on Pattern Analysis and Machine Intelligence*, *18*, 607–616.

[8] Hebrail, G., Hugueney, B., Lechevallier, Y., & Rossi, F. (2010). Exploratory analysis of functional data via clustering and optimal segmentation. *Neurocomputing*, *73*, 1125–1141.

[9] Jeong, Y., Jeong, M., & Omitaomu, O. (2011). Weighted dynamic time warping for time series classification. *Pattern Recognition*, *44*, 2231–2240.

[10] Kruskall, J., & Liberman, M. (1983). *The symmetric time warping algorithm: From continuous to discrete. In Time Warps, String Edits and Macromolecules.*. Addison-Wesley.

[11] Listgarten, J., Neal, R., Roweis, S., Puckrin, R., & Cutler, S. (2007). Bayesian detection of infrequent differences in sets of time series with shared structure. *Neural Information Processing Systems*, *19*.

[12] Navarro, G. (2001). A guided tour to approximate string matching. *ACM Computing Surveys*, *33*, 31–88.

[13] Paredes, R., & Vidal, E. (2006). Learning prototypes and distances: A prototype reduction technique based on nearest neighbor error minimization. *Pattern Recognition*, *39*, 180–188.

[14] Ramsay, J., & Li, X. (1998). Curve registration. *Journal of the Royal Statistical Society*, *B*, 351–363.

[15] Ratanamahatana, C. A., & Keogh, E. (2004). Making time-series classification more accurate using learned constraints. In *SIAM International Conference on Data Mining* (pp. 11–22.).

[16] Sakoe, H., & Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *26*, 43–49.

[17] Sankoff, D., & Kruskal, J. (1983). *Time warps, string edits, and macromolecules: the theory and practice of sequence comparison*. Addison-Wesley.

[18] Wartenberg, D. (1985). Multivariate spatial correlation: A method for exploratory geographical analysis. *Geographical Analysis*, *17*, 263–283.

[19] Xie, Y., & Wiltgen, B. (2010). Adaptive feature based dynamic time warping. *International Journal of Computer Science and Network Security*, *10*.

[20] Yu, D., Yu, X., Hu, Q., Liu, J., & Wu, A. (2011). Dynamic time warping constraint learning for large margin nearest neighbor classification. *Information Sciences*, *181*, 2787–2796.